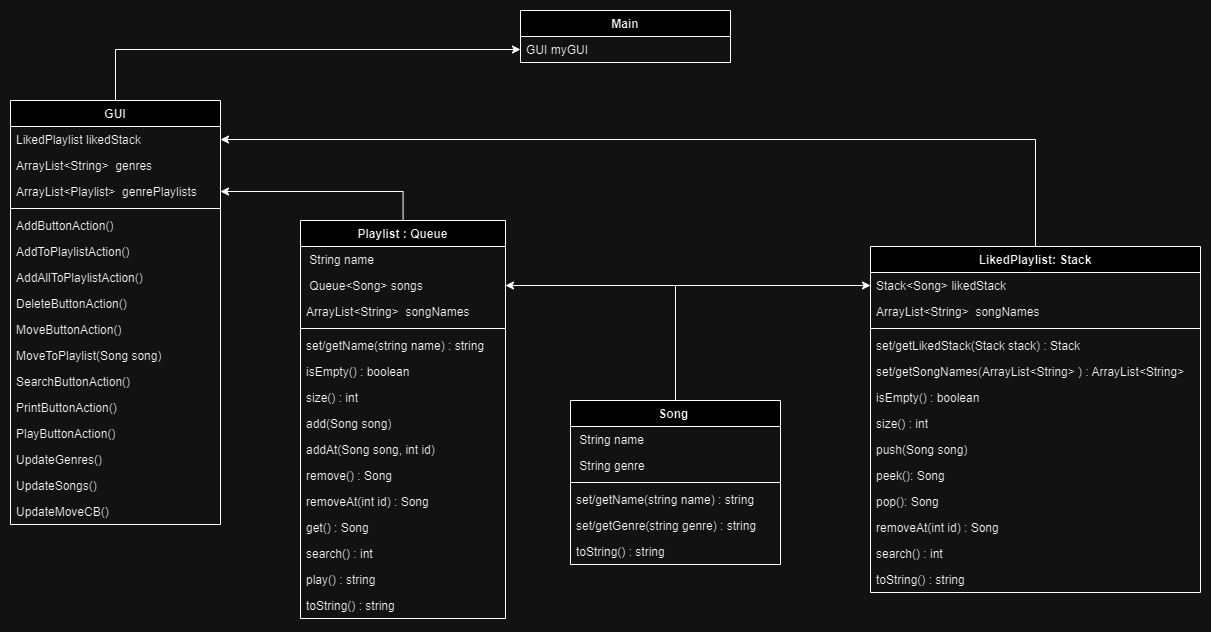
**Student: David Gevorkyan**

**Student Number: 21195242**

**Module: Data Structures and Algorithms**

**Lecturer: Hamilton Niculescu**

**Class Diagram**

****

**Main Class:** Main class stores a reference to the GUI class and invokes it on start.

**GUI Class:** Manages the GUI elements of the application and contains all the functions executed on button presses. Also contains a couple of helper functions that refresh the GUI elements based on programs state.

**Playlist Class:** Manages all songs that are sorted into playlists based on their genre. This class is based on the Queue advanced data type. This choice was made based on one of the functions of the class being playing songs in a queue and then adding it to the end of the queue. Also contains an Array List of the song names for the convenience of displaying the song names to the user without extracting all of them from the queue itself. Has a special toString() method for displaying contents.

**LikedPlaylist Class:** Is very similar to the playlist class except it is based on the Stack advanced data type. This decision was made based on one important limitation outlined in the brief, that being the limitation of only being able to add only the last song form the liked playlist to the other playlists. Like the Playlist class contains an Array List of the song names and a toString() function for displaying purposes.

**Song Class:** Manages individual songs with their variables and functions. Is the core class of the project and lies in base of both playlist classes. Also has its own specialized toString() function.

**GUI**

**A white rectangular object with blue border

Description automatically generated**

**Fields**

**Current Playlist:** The selected playlist by the user

**Current Song:** The selected song by the user

**Move To:** The selected position in the queue or stack to move the selected song to

**Search for song:** The text field used for searching a song in the playlist

**Name:** The text field used for inputting the song name

**Genre:** The text field used for inputting the song genre

**Repeat Amount:** The text field used for inputting the number of replays when the playlist is played.

**Buttons**

**Print, Delete, Move,** **Search:** Executes the function described by the buttons name

**Add to Playlist:** Adds the last song in the liked playlist to the corresponding genre playlist.

**Add all to Playlists:** Adds all the songs in the liked playlist to their corresponding genre playlists.

**Play current Playlist:** Plays the selected playlist ***Repeat Amount*** times

**Checkboxes**

**Auto Move to Playlists:** When set to true automatically adds the fresh created songs to their playlists.

**Text Fields**

**Console:** Is used for displaying useful information to the user

**Problem encountered**

The move functionality required being able to remove any element from the playlist and insert it back at the required index. The Queue data structure can`t remove or insert elements at a required index. Due to absence of any other solutions the queue is converted to Array List, the operation is carried on the List and then the resulting list is converted back to a Queue.

**GitHub Link:** https://github.com/Davo0416/MusicManager